Министерство науки и образования РФ

Федеральное государственное бюджетное учреждение

высшего образования

**«Тверской государственный технический университет»**

(ТвГТУ)

Кафедра программного обеспечения

**Курсовая работа**

по дисциплине: «Теория алгоритмов»

Тема: «Машина Поста (работа с серией программ на входе)»

|  |
| --- |
| Выполнил:  студент группы  Б.ПИН.РИС-21.06  Талакуцкая Анастасия |
| Проверила:  старший преподаватель кафедры ПО  Корнеева Е.И. |
|  |

Тверь 2023

**Содержание**

[**Введение** 3](#_Toc153355758)

[**Цели и задачи проекта:** 3](#_Toc153355759)

[**Теоретическая часть** 4](#_Toc153355760)

[**Теоретическая справка:** 4](#_Toc153355761)

[**Основные теоремы машины Поста включают:** 4](#_Toc153355762)

[**Сравнение с машиной Тьюринга**: 5](#_Toc153355763)

[**Алгоритм Машины Поста:** 5](#_Toc153355764)

[**Проектная часть** 6](#_Toc153355765)

[**Техническое задание:** 6](#_Toc153355766)

[**Цель проекта:** 6](#_Toc153355767)

[**Ключевые требования:** 6](#_Toc153355768)

[**Задачи проекта:** 6](#_Toc153355769)

[**Роли в проекте:** 6](#_Toc153355770)

[**Схема работы программы**: 7](#_Toc153355771)

[**Test Lead** 8](#_Toc153355772)

[**Заключение** 10](#_Toc153355773)

[**Выводы по работе:** 10](#_Toc153355774)

[**Выводы по своей роли:** 10](#_Toc153355775)

[**Список литературы** 11](#_Toc153355776)

# **Введение**

## **Цели и задачи проекта:**

Целью данного проекта является исследование и анализ машины Поста, а также определение областей ее применения в работе с серией программ на входе. Для достижения этой цели мы ставим следующие задачи:

* Изучение основных теорем машины Поста, включая теорему о существовании универсальной машины Поста.
* Исследование конкретных задач, которые можно решать с помощью машины Поста, такие как проверка на палиндром, обработка формальных языков и другие.
* Сравнение машины Поста с машиной Тьюринга с точки зрения вычислительной мощности, выразительности и применимости.

# **Теоретическая часть**

## **Теоретическая справка:**

Машина Поста состоит из каретки (или считывающей и записывающей головки) и разбитой на ячейки бесконечной в обе стороны ленты. Каждая ячейка ленты может находиться в 2 состояниях — быть либо пустой — 0, либо помеченной меткой 1. За такт работы машины каретка может сдвинуться на одну позицию влево или вправо, считать, изменить символ в своей текущей позиции.

Работа машины Поста определяется программой, состоящей из конечного числа строк. Для работы машины нужно задать программу и её начальное состояние (то есть состояние ленты и позицию каретки). Кареткой управляет программа, состоящая из пронумерованных не обязательно упорядоченных строк команд, если в каждой команде указана строка, на которую нужно перейти. Обычно принимается, что если в команде переход не указан, то переход происходит на следующую строку.

## **Основные теоремы машины Поста включают:**

* Теорема о существовании универсальной машины Поста: Существует машина Поста, которая может симулировать работу любой другой машины Поста, принимая ее описание и входные данные.
* Теорема о остановке: Невозможно написать программу на машине Поста, которая может точно определить, остановится ли данная машина Поста при выполнении данной программы.

Задачи, которые можно решать с помощью машины Поста, включают в себя проверку на палиндром, преобразование одного формального языка в другой, решение задачи о непустоте, и многие другие.

## **Сравнение с машиной Тьюринга**:

Машина Поста и машина Тьюринга обе являются моделями вычислений, но они имеют свои преимущества и недостатки:

***Преимущества машины Поста:***

* Она более интуитивно понятна и легче визуализируется, особенно при работе с последовательностями символов.
* Машина Поста более наглядно демонстрирует процессы, связанные с обработкой строковых данных.

***Недостатки машины Поста:***

* Она менее универсальна и ограничена в выразительности по сравнению с машиной Тьюринга.
* Невозможно определить остановку машины Поста для произвольной программы.

## **Алгоритм Машины Поста:**

1. Задать начальное состояние ленты и позицию каретки.

2. Задать программу, состоящую из конечного числа строк команд.

3. Установить текущую команду на первую строку программы.

4. Если текущая команда - ".", то остановить работу машины.

5. Если текущая команда - "V j", то поставить метку в текущей позиции и перейти к j-й строке программы.

6. Если текущая команда - "X j", то стереть метку в текущей позиции и перейти к j-й строке программы.

7. Если текущая команда - "← j", то сдвинуться влево на одну позицию и перейти к j-й строке программы.

8. Если текущая команда - "→ j", то сдвинуться вправо на одну позицию и перейти к j-й строке программы.

9. Если текущая команда - "? j1; j2", то если в текущей позиции нет метки, то перейти к j2-й строке программы, иначе перейти к j1-й строке программы.

10. Установить текущую команду на следующую строку программы.

11. Повторять шаги 4-10, пока не будет выполнена команда "!" или не будет достигнут максимальный лимит шагов.

# **Проектная часть**

## **Техническое задание:**

### **Цель проекта:**

Разработать программу для работы с серией программ на входе, используя Машину Поста.

### **Ключевые требования:**

- Разработка программы должна быть выполнена на языке Python;

- Программа должна иметь графический интерфейс пользователя (GUI);

- Программа должна содержать модульные тесты;

- Программа должна иметь

### **Задачи проекта:**

* Создать конечный автомат (машина Поста), который будет работать с введенной строкой и определять, является ли она палиндромом. Машина должна иметь состояния для чтения символов, перемещения влево и вправо по строке и принятия решения.
* Разработать интерфейс пользователя для ввода строки.
* Реализовать алгоритм работы машины Поста в программном коде.
* Провести тестирование программы на различных входных данных, включая палиндромы и не палиндромы, и убедиться в правильности определения.

### **Роли в проекте:**

1. Чесноков Дмитрий Сергеевич. Team Lead - Отвечает за общее руководство проектом, планирование и контроль выполнения работ, координацию команды.

2. Сорокин Евгений. Classes lead - Отвечает за разработку классов и методов для работы Машины Поста.

3. Леонов Алексей. GUI lead - Отвечает за разработку графического интерфейса пользователя.

4. Талакуцкая Анастасия. Test lead - Отвечает за разработку и проведение модульных тестов.

5. Лютый Максим. Docs lead - Отвечает за разработку документации.

## **Схема работы программы**:C:\Users\dcdim\AppData\Local\Microsoft\Windows\INetCache\Content.Word\diagram (2).png

## **Test Lead(добавить тестовые кейсы)**

import unittest

from post import PostMachine

class TestPostMachine(unittest.TestCase):

    def setUp(self):

        self.machine = PostMachine() #Создаём глобальный экземпляр класса Машины Поста

# Проверяется, что программа машины содержит только одну команду ">".

    def test\_add\_command(self):

        self.machine.add\_command(">") # вывается метод add\_command с аргументом ">", который добавляет команду ">" в программу машины.

        self.assertEqual(self.machine.program, [">"])

    def test\_right(self):

        self.machine.tape.right()  #Вызывается метод right у ленты машины, который перемещает каретку на одну позицию вправо.

        self.assertEqual(self.machine.tape.pos\_carriage, 1) #Проверяется, что позиция каретки равна 1.

    def test\_left(self): #Аналогично с тестом test\_right, только в другую сторону. Проверка, что будет -1

        self.machine.tape.left()

        self.assertEqual(self.machine.tape.pos\_carriage, -1)

if \_\_name\_\_ == "\_\_main\_\_":

    unittest.main()

Код представляет собой набор тестов для класса PostMachine. В каждом тесте создается экземпляр класса PostMachine, затем вызываются различные методы этого класса, и их результаты сравниваются с ожидаемыми значениями с помощью методов из библиотеки unittest.

Тест test\_add\_command: Проверяет, что метод add\_command() класса PostMachine корректно добавляет команду (в данном случае ">", предположительно команда для машины Поста) в программу машины.

**Заголовок:** Проверка добавления команды в программу машины Поста

**Предусловие**: Машина Поста self.machine инициализирована.

**Шаги:**

1.Вызвать метод add\_command(">") для машины self.machine. Этот метод добавляет команду ">" в программу машины.

**Ожидаемый результат:** После вызова метода add\_command(">"), программа машины должна содержать только одну команду ">", которая добавлена в программу.

Тест test\_right: Проверяет, что метод right() объекта tape (ленты) класса PostMachine корректно перемещает каретку на одну позицию вправо.

**Заголовок**: Проверка перемещения каретки на ленте машины Поста вправо

Предусловие: Машина Поста self.machine и лента self.machine.tape инициализированы.

**Шаги:**

* Вызвать метод right() для ленты машины self.machine.tape. Этот метод перемещает каретку на одну позицию вправо.
* Ожидаемый результат:

Ожидаемый результат: После вызова метода right(), позиция каретки self.machine.tape.pos\_carriage должна равняться 1, что указывает на перемещение каретки на одну позицию вправо.

Тест test\_left: Аналогично test\_right, только проверяет перемещение каретки на одну позицию влево.

**Заголовок:** Проверка перемещения каретки на ленте машины Поста влево

**Предусловие:** Машина Поста self.machine и лента self.machine.tape инициализированы.

**Шаги:**

* Вызвать метод left() для ленты машины self.machine.tape. Этот метод перемещает каретку на одну позицию влево.
* Ожидаемый результат:

**Ожидаемый результат:** после вызова метода left(), позиция каретки self.machine.tape.pos\_carriage должна равняться -1, что указывает на перемещение каретки на одну позицию влево.

Тест test\_num: Убеждается, что метод num() объекта tape класса PostMachine записывает указанное значение (в данном случае 0) на текущую позицию ленты.

**Заголовок:** Проверка записи числа на ленту машины Поста

**Предусловие:** Машина Поста self.machine и лента self.machine.tape инициализированы.

**Шаги:**

* Вызвать метод num(0) для ленты машины self.machine.tape. Этот метод записывает значение 0 на текущую позицию ленты.
* Ожидаемый результат:

**Ожидаемый результат:** После вызова метода num(0), содержимое ленты, полученное с помощью метода self.machine.get\_tape(), должно равняться "0", что указывает на успешную запись числа на текущую позицию ленты.

Тест test\_checknum: Устанавливает значение 1 на текущей позиции ленты с помощью метода num(), а затем использует метод checknum() объекта tape, чтобы проверить, что значение на текущей позиции соответствует ожидаемому (равно 1).

**Заголовок:** Проверка чтения числа с ленты машины Поста

**Предусловие:** Машина Поста self.machine и лента self.machine.tape инициализированы.

**Шаги:**

* Вызвать метод num(1) для ленты машины self.machine.tape. Этот метод записывает значение 1 на текущую позицию ленты.
* Вызвать метод checknum() для ленты машины self.machine.tape. Этот метод возвращает значение на текущей позиции ленты.

**Ожидаемый результат:**

После вызова метода num(1), значение на текущей позиции ленты становится равным 1.

После вызова метода checknum(), возвращенное значение должно быть равно 1, что указывает на успешное чтение числа с текущей позиции ленты.

Затем, если код запущен как основная программа (if name == "main"), выполняется запуск всех тестов с помощью unittest.main().

# **Заключение**

В ходе выполнения данной работы была проведена исследовательская работа по машине Поста и ее применению для решения типовой задачи на проверку строки на палиндром. Работа включала в себя анализ основных теорем машины Поста, описание решения задачи, а также разработку программы, использующей машину Поста для данной задачи.

## **Выводы по работе:**

* Машина Поста - это мощный инструмент в теории вычислений, который может использоваться для решения разнообразных задач, связанных с обработкой строковых данных.
* Основные теоремы машины Поста, включая теорему о существовании универсальной машины Поста, имеют фундаментальное значение в теории вычислений.
* Программа, разработанная на машине Поста, может успешно определять, является ли введенная строка палиндромом. Это является примером эффективного применения машины Поста в практической задаче.

## **Выводы по своей роли:**

Моя роль (Test Lead) в выполнении этой работы заключалась в создании модульных тестов для тестирования машины Поста, а также тестирование различных задач Машины поста.

# **Список литературы**

Машина Поста // Википедия URL::

<https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%B0_%D0%9F%D0%BE%D1%81%D1%82%D0%B0>  
(дата обращения: 27.09.2023)

Программирование на машине Поста (leventov) // Хабр URL: <https://habr.com/ru/articles/111272/>  
(дата обращения: 05.10.2023)

Машина Поста тренажер для изучения универсального исполнителя (Поляков К.) // kpolyakov.spb.ru URL: <https://kpolyakov.spb.ru/prog/post.htm>

(дата обращения: 17.10.2023)

unittest - Unit testing framework // Python URL: <https://docs.python.org/3/library/unittest.html>  
(дата обращения: 16.10.2023)

Тест-Кейс // https://practicum.yandex.ru URL: <https://practicum.yandex.ru/blog/chto-takoe-test-keys-i-kak-ego-sostavit/#pravila-sostavleniya>

(дата обращения: 04.12.2023)